Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

Tipo de artículo: Artículo original
Temática: Tecnologías de bases de datos
Recibido: 10/04/2016 | Aceptado: 15/05/2016

Proceso de réplica de datos con Microsoft SQL Server para el Replicador de Datos Reko

Data replication process with Microsoft SQL Server for Reko Data Replicator

Nayibi Martín Peña[1*], Marcos Manuel Martín Mata[2], Richel Labrada Quiala[3], Gloria Raquel Leyva Jerez[1]

[1] Centro de Consultoría y Desarrollo de Arquitecturas Empresariales - Facultad 5 - Universidad de las Ciencias Informáticas, Cuba, Carretera San Antonio de los Baños Km 2½, La Lisa, La Habana, Cuba,
{nmartin, grleyva}@uci.cu

[2] Facultad Universitaria Municipal Mayarí, Holguín, Cuba, marcos59@nauta.cu

[3] Empresa de Tecnología de la Información para la Defensa, Cuba, Carretera San Antonio de los Baños Km 2½, La Lisa, La Habana, Cuba, rquiala@xetid.cu

[*] Autor para correspondencia: nmartin@uci.cu

Resumen

El Replicador de Datos Reko pretende cubrir las principales necesidades relacionadas con la distribución de datos entre los gestores de bases de datos más populares como PostgreSQL, MySQL y Oracle, pero no contaba con un mecanismo que resolviera la replicación de datos con Microsoft SQL Server. Cuando se deseaba replicar datos con Microsoft SQL Server haciendo uso del Replicador de Datos Reko se hacía necesaria la utilización de métodos tradicionales de distribución de datos, lo que causaba una gestión de la información lenta y posibles errores en la consistencia de los datos. Con el objetivo de desarrollar un dialecto para el Replicador de Datos Reko que permitiese la replicación de datos con Microsoft SQL Server se utilizaron como métodos científicos de investigación los métodos teóricos: "Análisis Histórico-Lógico" y "Analítico-Sintético". Además, se aplicó la metodología de desarrollo ágil Open Up que abarca todo el proceso de desarrollo del software con iteraciones pequeñas. El software ha sido diseñado e implementado en su totalidad usando herramientas Open Source y librerías de clases con licencias gratuitas, como: Visual Paradigm, Eclipse STS, Apache ActiveMQ, Apache Tomcat, entre otras. El dialecto de Microsoft SQL Server para Reko facilitará realizar configuraciones de réplica para este gestor, sincronizar, enviar y recibir datos con el gestor de base de datos Microsoft SQL Server y replicar datos desde Microsoft SQL Server hacia otros gestores, por lo que constituirá una solución madura y viable ante las necesidades de los sistemas que utilicen este replicador.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

Palabras clave: replicación de datos; dialecto; Microsoft SQL Server

Abstract

Reko Data Replicator intends to cover the principal needs related to data distribution between most popular databases managers as PostgreSQL, MySQL and Oracle, but did not have a mechanism to resolve data replication with Microsoft SQL Server. When you want to replicate data using the database manager Microsoft SQL Server required the use of traditional methods of data distribution, what caused a slow information management and possible errors in the data consistency. With the objective of developing a dialect for Reko Data Replicator that allow data replication with Microsoft SQL Server were used scientific methods of investigation, they were the theoretical methods: "Logical-Historical Analysis" and "Analytic-Synthetic". Also applied the agile development methodology Open Up spanning the entire software development process with small iterations. The software has been designed and implemented entirely using Open Source tools and class libraries with free licenses such as: Visual Paradigm, Eclipse STS, Apache ActiveMQ and Apache Tomcat. The Microsoft SQL Server dialect for Reko facilitate replication configurations for this manager, sync, send and receive data in Microsoft SQL Server manager and replicate data from Microsoft SQL Server to other managers, so it will be a mature and viable solution to the needs of systems using this replicator.

Keywords: *data replication; dialect; Microsoft SQL Server*

Introducción

El constante manejo de la información y el perfeccionamiento de las comunicaciones a nivel mundial han conducido a establecer sistemas de información que faciliten almacenar datos de forma masiva, dando lugar al surgimiento de los sistemas de bases de datos. En sus inicios los datos se almacenaban de forma centralizada, pero el incremento de la información y la necesidad de que ordenadores ubicados geográficamente distantes pudiesen compartir datos impulsó vertiginosamente la utilización de sistemas distribuidos. La tendencia al desarrollo de estos sistemas ha propiciado que las bases de datos que se encuentran en ordenadores distantes, favorecidas por la utilización de las redes de comunicación, mantengan la actualización y sincronización de los datos a través de los sistemas de réplica. El proceso de replicación permite el copiado y distribución de objetos de una base de datos hacia otra, lo que contribuye a la coherencia y consistencia de la información.

Precisamente en la Universidad de las Ciencias Informáticas (UCI), se encuentra ubicado el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), en el cual se desarrolla el Replicador de Datos Reko. Este software surge a causa de la necesidad que se presentó en la distribución de datos en el Sistema de Gestión Penitenciario Venezolano (SIGEP) en el año 2007. Reko procura cubrir las principales necesidades relacionadas con

la distribución de datos entre los gestores de bases de datos más populares como PostgreSQL, MySQL y Oracle en la protección, recuperación, sincronización de datos, transferencia de datos entre diversas localizaciones y la centralización de la información en una única localización. Está siendo utilizado por varios sistemas tanto a nivel nacional como internacional entre los que se encuentran el sistema DATAFAR de las Fuerzas Armadas Revolucionarias de la República de Cuba, el proyecto de Informatización de Tribunales, el sistema de Fiscalía y actualmente está siendo desplegado en el Sistema de Gestión Hospitalaria de PDVSA.

Igualmente, el Sistema Nacional de Gestión de Ingreso a la Educación Superior ha hecho uso de la herramienta en una etapa inicial. La Universidad de La Habana desea utilizar Reko para realizar la replicación de datos en el sistema antes mencionado; para ello necesita consultar varias bases de datos construidas con el gestor de base de datos Microsoft SQL Server y que se encuentran ubicadas en otros sistemas legados, para luego replicar los datos hacia su base de datos en MySQL. Actualmente Reko no permite la replicación de datos utilizando el gestor de base de datos Microsoft SQL Server por lo cual se reduce su utilización en los sistemas que requieren el empleo de este gestor como es el caso del Sistema Nacional de Gestión de Ingreso a la Educación Superior. Para poder efectuar la réplica con Microsoft SQL Server haciendo uso del Replicador de Datos Reko se recurren a métodos tradicionales, como la copia de *backups* y *scripts* hacia las distintas bases de datos, lo que causa una gestión de la información lenta y los datos corren el riesgo de poseer errores de coherencia y consistencia.

El software Microsoft SQL Server cuenta con un mecanismo de replicación que pudiera ser utilizado para copiar y distribuir datos entre las bases de datos, pero el mismo tiene como inconveniente que de los gestores de base de datos a los cuales Reko brinda soporte de réplica, sólo es compatible con Oracle. Además, posee un alto costo tanto monetario como de recursos, lo que implicaría un mayor gasto económico y no se resolvería totalmente la replicación de datos.

Para dar solución al problema planteado se define como objetivo de la investigación desarrollar un dialecto de Microsoft SQL Server para el Replicador de Datos Reko que permita realizar configuraciones de réplica para este gestor, además de sincronizar, enviar y recibir datos con el gestor de bases de datos Microsoft SQL Server.

Una vez definida la problemática se realizó un estudio de los gestores de bases de datos a los cuales Reko brinda opción de réplica de datos: PostgreSQL, MySQL y Oracle, además de Microsoft SQL Server. Se realizó una comparación detallada y valorativa entre las principales sintaxis, tipos de datos y consultas SQL de estos gestores y que luego iban a ser utilizadas para el desarrollo del dialecto de Microsoft SQL Server. Entre las sintaxis más

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

importantes que fueron estudiadas se encuentran: crear y eliminar disparadores (*triggers*), habilitar y deshabilitar disparadores, habilitar y deshabilitar llaves foráneas, crear, actualizar y eliminar esquemas, tablas y otras estructuras.

Materiales y métodos o Metodología computacional

Proceso de replicación de datos en Microsoft SQL Server

La replicación en Microsoft SQL Server radica en el transporte de datos entre dos o más instancias de servidores, y para ello brinda un conjunto de soluciones que permite la copia, distribución y modificación de los datos, además incluye varios métodos y opciones para el diseño como: implementación, supervisión y administración de la replicación, con el fin de mantener la coherencia y seguridad de los datos (MICROSOFT DEVELOPER NETWORK, 2015a).

Microsoft SQL Server realiza la réplica de los datos basándose en una metáfora de la industria de la publicación, por lo que representa los componentes y procesos mediante una topología de replicación, que se compone, básicamente, del publicador, el distribuidor, los suscriptores, las publicaciones, los artículos y las suscripciones (MICROSOFT SQL SERVER, 2015a).

Una topología de replicación define la relación entre los servidores y las copias de los datos, y determina cómo se desarrollará el flujo de la información (APINA and ROBLES, 2011). En la Figura 1, se muestra información general acerca de los componentes y procesos que participan en la replicación.

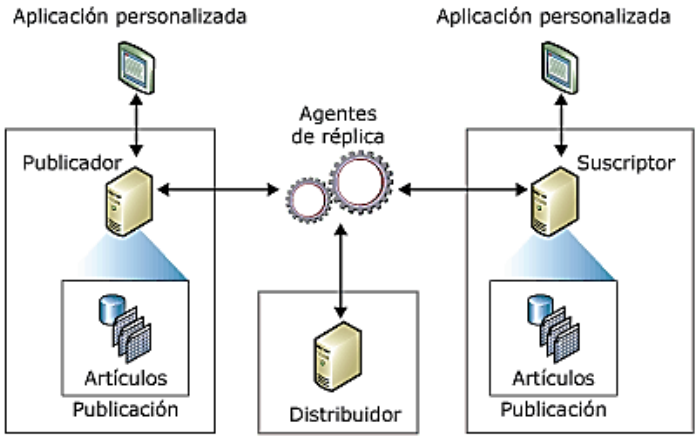

Figura 1. Componentes y procesos que participan en la replicación en Microsoft SQL Server

- Publicador: es una instancia de base de datos que permite que los datos estén disponibles para otras ubicaciones a través de la replicación.
- Distribuidor: es una instancia de base de datos que funciona como almacén para datos específicos de replicación asociados con uno o más publicadores. Cada publicador está asociado con una sola base de datos en el distribuidor.
- Suscriptor: es una instancia de base de datos que recibe datos replicados. El suscriptor también puede devolver los datos modificados al publicador o volver a publicar los datos en otros suscriptores.
- Artículo: identifica un objeto de base de datos incluido en una publicación, como, por ejemplo: tablas, vistas, procedimientos almacenados y otros objetos.
- Publicación: es un conjunto de uno o más artículos de una base de datos. La publicación permite especificar un conjunto de objetos y datos de bases de datos que se replican como una unidad.
- Suscripción: es una solicitud de una copia de una publicación que se entrega a un suscriptor. La suscripción define qué publicación se recibirá, dónde y cuándo (MICROSOFT DEVELOPER NETWORK, 2015b).
- Agentes de Replicación: son programas independientes para realizar las tareas asociadas con el seguimiento de los cambios y la distribución de los datos. Existen varios agentes de replicación como: el agente SQL Server, el agente de instantáneas, el agente de registro del LOG, el agente de distribución, el agente de mezcla, el agente de lectura de cola y los trabajos de mantenimiento de la replicación (MICROSOFT SQL SERVER, 2015b).

Inconvenientes de Microsoft SQL Server

Como consecuencia de contar con una robusta tecnología, Microsoft SQL Server posee un alto precio económico, en dependencia de las opciones que se requiera del gestor de base de datos, llegando a costar hasta 25 000 dólares. Se puede agregar además que la replicación con Microsoft SQL Server está definida sólo para sistemas operativos de Microsoft, e incluso en ocasiones es incompatible con algunos entornos del mismo, por lo que es imposible trabajar en cualquier tipo de ambiente de desarrollo. Otro de los inconvenientes de Microsoft SQL Server es que requiere de una gran cantidad de memoria RAM para poder instalarlo y utilizarlo, limitando su usabilidad.

Sistema Gestor de Base de Datos Microsoft SQL Server

Para el desarrollo del dialecto de Microsoft SQL Server para el Replicador de Datos Reko fue necesario realizar un estudio de las principales sintaxis de este gestor de bases de datos. Microsoft SQL Server es un SGBD (Sistema

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

Gestor de Bases de Datos) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Los tipos de datos de Microsoft SQL Server son compatibles con la mayoría de tipos de datos de Oracle, DB2 y MySQL, es decir, que se pueden utilizar los mismos formatos para crear y almacenar datos en las diferentes bases de datos existentes (ALOMALIZA, 2008). Los principales tipos de datos que soporta la plataforma Microsoft SQL Server se muestran en la Tabla 1:

Tabla 1. Tipos de datos en Microsoft SQL Server

Tipos de Datos en Microsoft SQL Server		
Alfanuméricos	**Numéricos**	**Fecha**
char	tinyint	smalldatetime
varchar	smallint	datetime
binary	int	timestamp
varbinary	bigint	date
text	float	time
image	money	datetime2
nchar	numeric	datetimeoffset
nvarchar	real	
ntext	decimal	
	bit	
	smallmoney	

En Microsoft SQL Server los desencadenadores o disparadores (*triggers*) se ejecutan cuando un usuario intenta modificar datos mediante un evento DML a través de las instrucciones *INSERT*, *UPDATE* o *DELETE* de una tabla o vista. Estos desencadenadores se activan cuando se desencadena cualquier evento válido, con independencia de que las filas de la tabla se vean o no afectadas. Para crear *triggers* se utiliza la siguiente sintaxis:

CREATE TRIGGER [*schema.*] *trigger*
ON *table*
{FOR | AFTER | INSTEAD OF}
{ [INSERT] [,] [UPDATE] [,] [DELETE] }
AS *sql_sentence* (1)

donde:

schema: es el nombre del esquema al que pertenece un desencadenador. Este campo es opcional.
trigger: es el nombre del disparador (*trigger*).
table: es la tabla o vista en que se ejecuta el desencadenador.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

sql_sentence: son las condiciones y acciones del desencadenador. (MICROSOFT DEVELOPER NETWORK, 2015c)

Para eliminar triggers que se encuentran en la base de datos se utiliza la siguiente sintaxis:

DROP TRIGGER [*schema.*] *trigger* (2)

donde:

schema: es el nombre del esquema al que pertenece un desencadenador. Este campo es opcional.

trigger: es el nombre del disparador (*trigger*). (MICROSOFT DEVELOPER NETWORK, 2015d)

Para habilitar y deshabilitar los disparadores o desencadenadores se utilizan las siguientes sintaxis:

ENABLE TRIGGER [*schema.*] *trigger*
ON *table* (3)

DISABLE TRIGGER [*schema.*] *trigger*
ON *table* (4)

donde:

schema: es el nombre del esquema al que pertenece un desencadenador. Este campo es opcional.

trigger: es el nombre del disparador (*trigger*). MICROSOFT DEVELOPER NETWORK (2015e, 2015f)

Para habilitar y deshabilitar llaves foráneas en Microsoft SQL Server se utilizan las siguientes sintaxis:

ALTER TABLE [*schema.*]*table*
CHECK CONSTRAINT *foreign_key* (5)

ALTER TABLE [*schema.*]*table*
NOCHECK CONSTRAINT *foreign_key* (6)

donde:

schema: es el nombre del esquema al que pertenece la llave foránea. Este campo es opcional.

table: es la tabla o vista a la que pertenece la llave foránea.
foreign_key: es el nombre de la llave foránea. (CHERRY,2011)

Características del Replicador de Datos Reko

El Replicador de Datos Reko es un software de réplica de datos que ha sido implementado en la Universidad de las Ciencias Informáticas como respuesta a la necesidad de mantener actualizadas un conjunto de bases de datos y pretende cubrir las principales necesidades relacionadas con la distribución de datos entre los gestores más populares. Ha sido diseñado para permitir la réplica y sincronización de datos con conexión y sin conexión, la selección de los datos a replicar y la definición de filtros de replicación, la réplica de datos entre bases de datos con estructuras heterogéneas, y entre gestores heterogéneos.

Entre las principales funcionalidades que brinda el Replicador de Datos Reko se encuentran:

- Soporte para réplica de datos en ambientes con conexión y sin conexión
- Soporte para réplica entre bases de datos con diferentes estructuras
- Soporte para réplica de ficheros externos a la base de datos
- Soporte para la sincronización de datos en ambientes con conexión y sin conexión
- Selección de los datos de réplica ajustada por filtros
- Soporte para réplica de datos entre gestores diferentes (Postgres-Oracle-MySQL)
- Soporte para resolución de conflictos automática y manualmente
- Monitoreo en tiempo real de los datos de réplica
- Réplica de datos de gran tamaño con capacidad de resumir el envío y el recibo de los mismos en caso de desconexión
- Soporte para la programación del momento de captura y envío de los datos de réplica

Reko proporciona características y beneficios que aportan seguridad al cliente, destacándose que el software es independiente de la plataforma, replica ficheros de gran tamaño, replica datos en sistemas heterogéneos donde prevalecen diferentes gestores de base de datos, detecta errores de conexión, la administración y configuración de la réplica se realiza a través de una interfaz web por lo que es administrable vía remota usando un navegador, entre otras características que hacen de Reko un software maduro y una solución robusta ante el alto conjunto de escenarios de replicación que soporta, contando además con una comunidad de desarrollo que lo respalda (COMPANIONI, 2012).

Si bien el Replicador de Datos Reko cuenta con un conjunto de funcionalidades que hacen de él un software robusto y seguro ante las necesidades de réplica a nivel mundial, aún no cuenta con un mecanismo que resuelva la replicación de datos con el gestor de bases de datos Microsoft SQL Server, por lo que se reduce su utilización en sistemas que requieran la utilización de este gestor para almacenar información.

Resultados y discusión

El Replicador de Datos Reko permitía realizar configuraciones de réplica, capturar y enviar datos, y sincronizar los datos, en ambientes con conexión y sin conexión utilizando los gestores de bases de datos Oracle, MySQL y PostgreSQL. Una vez que se ejecuta el software, el mismo se conecta a la base de datos a partir de las propiedades definidas en un fichero denominado *replication.properties*, luego al realizar la configuración de réplica y la captura de los datos a replicar, se obtiene un dialecto determinado en dependencia de las características de cada gestor de base de datos.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

Cuando se deseaba efectuar la replicación de datos en el gestor de bases de datos Microsoft SQL Server, Reko no era capaz de reconocer los cambios efectuados en la base de datos debido a que, aunque algunas de las sentencias de Microsoft SQL Server son compatibles con otros gestores, no le era posible utilizar los dialectos de estos para construir las consultas. Una vez analizadas las limitantes antes expuestas se hizo necesario brindarle al Replicador de Datos Reko un conjunto de funcionalidades a través del dialecto de Microsoft SQL Server, que permitan construir consultas para luego ejecutarlas en la base de datos confeccionada en el gestor de base de datos Microsoft SQL Server.

Descripción de la arquitectura del Replicador de Datos Reko

Reko presenta una arquitectura que puede definirse como basada en componentes, debido a que todas sus partes encapsulan un conjunto de comportamientos que pueden ser reemplazados por otros. Los principales componentes presentes en el software son:

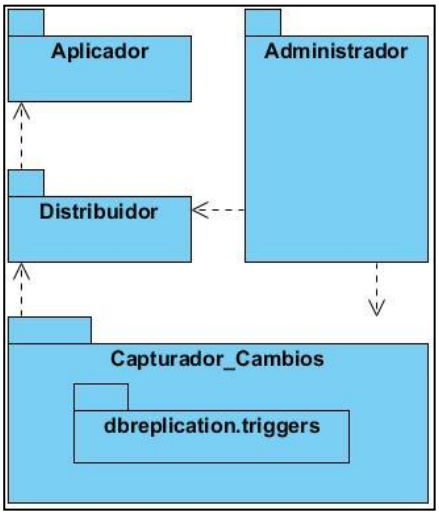

Figura 2. Vista de los Principales Componentes
de Reko, agrupados por paquetes

- Capturador_Cambios: encargado de capturar los cambios que se realizan en la base de datos y entregarlos al distribuidor.
- Distribuidor: determina el destino de cada cambio realizado en la base de datos, los envía y se responsabiliza de su llegada.
- Aplicador: ejecuta en la base de datos los cambios que son enviados hacia él desde otro nodo de réplica.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

- Administración: permite realizar las configuraciones principales del software como el registro de nodos, configuración de las tablas a replicar y el monitoreo del funcionamiento.

Independientemente de lo antes expuesto, el componente Administración responde a un modelo multicapas, donde cada capa tiene funcionalidades y objetivos precisos, así su implementación se encuentra desacoplada de la programación de cualquier otra y la comunicación con una capa inferior ocurre a través de interfaces. Además de estar separadas lógica y estructuralmente, las capas se encuentran separadas de manera física. Para realizar la replicación en el software se efectúa la construcción de consultas SQL que posteriormente se ejecutarán en la base de datos. Para lograr replicar con el gestor de bases de datos Microsoft SQL Server es necesario realizar modificaciones en el paquete *triggers*, correspondiente al componente Capturador_Cambios, pues, hasta el momento, sólo se tenían definidos los dialectos para PostgreSQL, MySQL y Oracle. En la Figura 2, se muestran los principales componentes del Replicador de Datos Reko.

Cambios realizados en el paquete *trigger*

Para dar solución a la problemática planteada y utilizando la metodología de desarrollo OpenUp se definieron requisitos funcionales con el fin de guiar el proceso de desarrollo del software, entre los que se encuentran: configurar los datos de la base de datos, configurar las estructuras de las tablas a replicar y capturar los datos a replicar.

Inicialmente, cuando se ejecuta el Replicador de Datos Reko, la clase *CustomBasicDataSource* construye el URL a partir de las propiedades de la base de datos definidas en el fichero *replication.properties* para que luego el sistema pueda conectarse a la base de datos. Una vez ejecutado el replicador es necesario crear una configuración de réplica que contendrá las tablas que van a ser replicadas y el tipo de réplica que desea el usuario, ya sea por inserción, actualización o eliminación.

Cuando se desea guardar esta configuración de réplica, se realiza una petición a la clase *TriggerReplicationManager*, perteneciente al paquete *triggers*, de crear las estructuras auxiliares en la base de datos para poder almacenar los datos que se van a replicar, la clase *TriggerReplicationManager* crea la estructura de las tablas replicables y el sistema, a partir del dialecto de Microsoft SQL Server, crea las tablas para almacenar los nombres y registros de las tablas con datos a replicar, crea en la base de datos las funciones para habilitar y deshabilitar llaves foráneas, crea en la base de datos una tabla espejo para cada una de las tablas con datos a replicar y crea en la base de datos los disparadores de *Insert*, *Update* y *Delete* para cada una de las tablas con datos a replicar. Todas estas acciones que se ejecutan en la base de datos son creadas a través de consultas definidas en el dialecto de Microsoft SQL Server (*SQLServerDialect*).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

Luego de que la configuración de réplica es guardada, se puede dar inicio a la captura de los cambios en los datos que se encuentran en la base de datos de Microsoft SQL Server. Para ello se realiza una petición a la clase *TriggerReplicationManager* de iniciar la captura de los datos que se van replicar y el sistema se encarga primeramente de verificar si se puede comenzar a replicar, se obtiene la cantidad de datos a replicar, se obtiene la acción replicable para cada grupo replicable, se adiciona la acción replicable al grupo replicable y por último se adicionan los grupos replicables a la lista de elementos a replicar.

Herramientas Utilizadas

Desde sus inicios Reko ha sido diseñado e implementado en su totalidad usando herramientas *Open Source* y librerías de clases con licencias gratuitas, lo que permite que el software pueda ser desplegado en cualquier sistema operativo. Estas herramientas fueron utilizadas también en el desarrollo del dialecto de Microsoft SQL Server para lograr una total compatibilidad con las demás funcionalidades existentes en el software. Con el fin de guiar el proceso de desarrollo de software se utilizó la metodología Open UP, como lenguaje de modelado UML 2.0 y la herramienta CASE[1] Visual Paradigm 8.0, además para llevar a cabo el desarrollo se utilizó como lenguaje de programación Java[2], el entorno de desarrollo integrado Eclipse STS[3] 2.3.2 y el framework[4] Spring 2.5.0. Se hizo uso también de Apache Tomcat 7.0.23 como servidor web, Apache ActiveMQ 5.4.3 como servidor de mensajería y el framework JMS para la gestión de mensajes. Para el desarrollo de pruebas unitarias se utilizó el framework JUnit 4.0 y como gestor de base de datos Microsoft SQL Server 2008.

Evaluación del diseño aplicando métricas de software

Para evaluar la calidad del diseño del dialecto de Microsoft SQL Server para el Replicador de Datos Reko fueron seleccionadas las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC).

Los resultados obtenidos al aplicar las métricas de software, arrojaron que el diseño propuesto tiene una calidad aceptable, teniendo en cuenta que el 66% de las clases poseen una cantidad de procedimientos menor o igual al promedio general de 17,5, esto conlleva a que las evaluaciones sean positivas en los atributos de calidad involucrados. Además, el 66% de las clases tienen una baja responsabilidad ya que este atributo se distribuyó equitativamente entre todas las clases del sistema. Esta característica permite que en caso de fallos como la responsabilidad está distribuida

[1] Ingeniería de Software Asistida por Computación

[2] Lenguaje de programación orientado a objetos

[3] Spring Tool Suite, por sus siglas en inglés.

[4] Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

de forma equilibrada ningún componente sea demasiado crítico como para dejar fuera de servicio el sistema. El 66% de las clases tienen una baja complejidad, este atributo está distribuido equitativamente. Esta característica permite mejorar el mantenimiento y soporte de las clases. El diseño de la solución es eficiente ya que el 66 % de las clases tienen un alto grado de reutilización. En la Figura 3, se muestran los resultados de la evaluación de la métrica TOC para el atributo de calidad Complejidad de implementación.

Los resultados obtenidos luego de aplicar las métricas RC arrojan que el diseño propuesto tiene una calidad aceptable, teniendo en cuenta que el 72% de las clases poseen una cantidad de relaciones de uso menor o igual al promedio general de 2,8, esto conlleva a que las evaluaciones sean positivas en los atributos de calidad involucrados. Además, se evidencia un diseño eficiente al quedar reflejado que un 60% de las clases cuentan con un bajo acoplamiento. Se asegura la eficiencia de la arquitectura del sistema al evidenciarse que un 80% de las clases posee una baja complejidad de mantenimiento. El 80% de las clases poseen una alta reutilización lo que es un factor fundamental que debe ser tenido en cuenta en el desarrollo de software. Y por último un 80% de las clases posee baja cantidad de pruebas, lo que representa valores favorables para el diseño realizado. En la Figura 4, se muestran los resultados de la evaluación de la métrica RC para el atributo de calidad Acoplamiento.

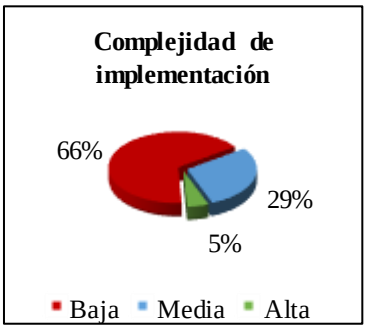

Figura 2. Resultados de la evaluación de la métrica TOC para el atributo de calidad Complejidad de implementación

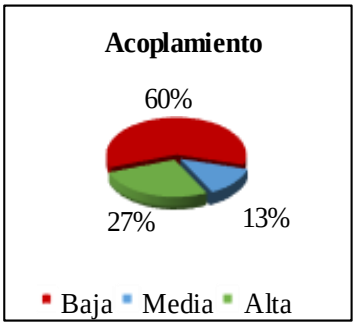

Figura 2. Resultados de la evaluación de la métrica RC para el atributo de calidad Acoplamiento

Una vez elaborada la matriz de inferencia de indicadores de calidad, todos los resultados obtenidos fueron positivos por lo que el promedio en todos los indicadores de calidad fue igual a 1. La Figura 5, muestra la gráfica de los resultados obtenidos de los atributos de calidad evaluados en las métricas aplicadas anteriormente, donde todos los atributos de calidad mantienen un buen comportamiento.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 171-185

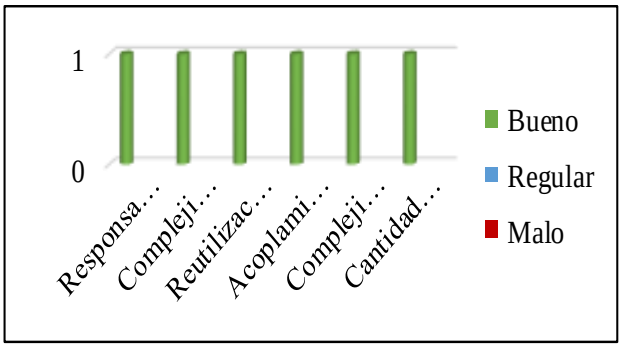

Figura. 3: Resultados obtenidos de la evaluación de los atributos de calidad

Pruebas de Software

Para comprobar los métodos de la clase *SqlServerDialect* se realizaron pruebas unitarias utilizando la técnica de caja blanca, las cuales posibilitaron comprobar cada uno de los métodos por separado y definir si el software estaba listo y correctamente terminado.

Las pruebas unitarias permitieron verificar el comportamiento de la clase *SqlServerDialect* y sus métodos. Mediante estas pruebas fue posible comparar la respuesta que debería obtenerse de un método con lo que realmente se adquiría de él. Actualmente existen varias herramientas que facilitan la elaboración y desarrollo de casos de pruebas, además de darle seguimiento a los errores, un ejemplo de ello es JUnit utilizada para realizar pruebas unitarias de aplicaciones Java.

Para la ejecución de estas pruebas fue necesario diseñar una clase denominada *SqlServerDialectTest* la cual contenía todos los casos de pruebas en correspondencia con los métodos de la clase *SqlServerDialect*. En total fueron elaborados 13 casos de pruebas que fueron ejecutados de forma automática cuando se realizaba la captura de los datos. En el caso de los métodos desarrollados para la configuración de réplica las pruebas se ejecutaban correctamente si se creaba una nueva configuración de réplica o existía alguna almacenada.

En una primera etapa de pruebas unitarias se obtuvo que, de las 13 pruebas, 4 fueron fallidas debido a errores de bajo impacto en la construcción de las consultas elaboradas en el dialecto. En una segunda etapa se pudieron resolver estos problemas, por lo que el resultado de las pruebas fue satisfactorio.

Además, fueron diseñados 7 casos de pruebas del sistema que fueron ejecutadas en tres etapas. En una primera etapa se detectaron errores de impacto medio en 4 de los requisitos funcionales que se estaban validando. En una segunda

etapa se lograron corregir 2 de estos errores. Y finalmente en una tercera etapa se logró que las pruebas para validar los requisitos se ejecutaran correctamente, emitiendo resultados satisfactorios.

Reko 3.1, con las nuevas funcionalidades de réplica de datos utilizando Microsoft SQL Server, fue liberada por CALISOFT (Centro de Calidad de Software) este año y actualmente se está culminando el proceso de registro de software por el mismo centro, donde se corroboró la correcta implementación de estas nuevas funcionalidades.

A partir de los resultados obtenidos se puede proponer la utilización de Reko para garantizar la replicación de datos en entornos de sistemas con necesidad de réplica en bases de datos Microsoft SQL Server ya que es una herramienta poderosa, multiplataforma y cubre las principales consultas SQL para lograr la réplica en gestores de bases de datos Microsoft SQL Server.

Conclusiones

Con el desarrollo del dialecto de Microsoft SQL Server para el Replicador de Datos Reko se agregaron funcionalidades que dan la posibilidad a los usuarios que utilizan este replicador realizar configuraciones de réplica para Microsoft SQL Server, sincronizar, enviar y recibir datos en ambientes con conexión y sin conexión utilizando el gestor de bases de datos Microsoft SQL Server.

Es una solución de gran impacto en la sustitución de importaciones pues fue desarrollada utilizando tecnologías modernas y libres. Además, fue validada en un ambiente de réplica real, por lo que constituye una solución robusta y de alta calidad.

Se recomienda añadir al dialecto la posibilidad de construir consultas SQL para la réplica de estructuras.

Referencias

ALOMALIZA, W. C. Estudio comparativo de mecanismos de migración automática de datos a plataformas heterogéneas aplicado al control biométrico de la ESPOCH. Tesis de Grado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2008.

APINA, A. E.; ROBLES, R. F. Estudio de agentes en servicios web para la replicación de bases de datos heterogéneas caso práctico: Aplicación Satélite Contilibro. Tesis de Grado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2011.

CHERRY, D. Microsoft SQL Server: La solución de secuenciación., [En línea]. Ingeniería Informática, 2011, [Consultado el: 16 de octubre de 2015]. Disponible en: [http://technet.microsoft.com/es-es/magazine/hh407114.aspx].

COMPANIONI, Y. Reko: Una solución de réplica para sistemas de bases de datos relacionales distribuidos. En: IV Simposio Informática y Comunidad. La Habana: 2012, 10 p.

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Replicación de SQL Server, 2015a. [Consultado el: 30 de septiembre de 2015]. Disponible en: [http://msdn.microsoft.com/es-es/library/ms151198.aspx].

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Información general del modelo de publicación de replicación, 2015b. [Consultado el: 5 de octubre de 2015] Disponible en: [http://msdn.microsoft.com/es-es/library/ms152567.aspx].

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Create trigger (Transact-SQL), 2015c. [Consultado el: 12 de octubre de 2015]. Disponible en: [http://msdn.microsoft.com/es-es/library/ms189799.aspx].

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Drop trigger (Transact-SQL), 2015d. [Consultado el: 15 de octubre de 2015]. Disponible en: [http://msdn.microsoft.com/es-es/library/ms173497.aspx].

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Enable trigger (Transact-SQL), 2015e. [Consultado el: 15 de octubre de 2015]. Disponible en: [http://msdn.microsoft.com/es-es/library/ms182706.aspx].

MICROSOFT DEVELOPER NETWORK. Microsoft SQL Server. [En línea]. Disable trigger (Transact-SQL), 2015f. [Consultado el: 15 de octubre de 2015]. Disponible en: [http://msdn.microsoft.com/es-es/library/ms189748.aspx].

MICROSOFT SQL SERVER. Microsoft SQL Server. [En línea]. Introducción (replicación), 2015a. [Consultado el: 2 de octubre de 2015] Disponible en: [http://technet.microsoft.com/es-es/library/bb500346%28v=sql.105%29.aspx].

MICROSOFT SQL SERVER. Microsoft SQL Server. [En línea]. Agentes de replicación, 2015b. [Consultado el: 7 de octubre de 2015]. Disponible en: [http://technet.microsoft.com/es-es/library/bb522755.aspx].